Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: OrderMakanan.java

Suatu hari, Kebin membuka bisnis F&B yang ia jalankan seorang diri, mulai dari menjadi koki, kasir, hingga mengantar pesanan. Awalnya semua berjalan lancar, namun seiring waktu jumlah Nimons yang memesan makanan semakin banyak. Kebin pun mulai kewalahan mencatat pesanan, menghitung total harga, dan memastikan setiap Nimons mendapat makanan sesuai pesanannya.

Agar bisnisnya tidak kacau, Kebin membutuhkan bantuan Anda untuk membuat sistem pemesanan makanan sederhana. Sistem ini akan dibuat menggunakan konsep Object-Oriented Programming (OOP), dengan membuat sebuah kelas bernama OrderMakanan yang nantinya akan dipakai oleh program utama untuk mengelola data pesanan setiap pelanggan.

Setiap pesanan (order) harus menyimpan tiga informasi penting:

1. Nama makanan yang dipesan.
2. Banyaknya makanan yang dipesan.
3. Harga satuan makanan tersebut.

Kebin juga butuh agar sistemnya bisa:

1. Mengubah jumlah pesanan jika ada penambahan atau pengurangan.
2. Menghitung total harga dari pesanan tertentu.

Petunjuk

1. Buat kelas bernama `OrderMakanan` yang berisi data nama, banyaknya makanan, dan harga satuannya.
2. Buat konstruktor yang menerima semua atribut sebagai parameter.
3. Buat getter dan setter untuk setiap atribut.
4. Buat method tambahan:
   - `increasecountMakanan(int tambahan)` → menambah jumlah pesanan jika nilai tambahan lebih dari 0.
   - `decreasecountMakanan(int pengurangan)` → mengurangi jumlah pesanan jika nilai pengurangan lebih dari 0.
   - `getTotalHarga()` → menghitung total harga pesanan
5. Lengkapi dan submit file **`OrderMakanan.java`** saja.

Java 8

OrderMakanan.java

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.06 sec, 28.93 MB
2	10	Accepted	0.06 sec, 28.93 MB

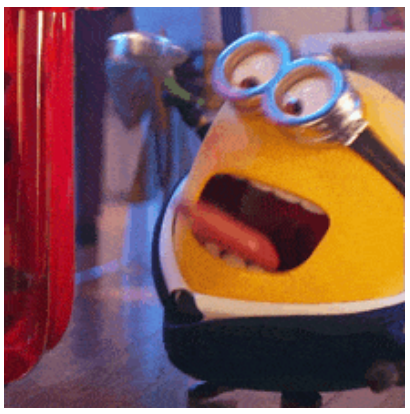
Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: Pembeli.zip



Setelah sistem `OrderMakanan` yang Anda buat berjalan, antrian Nimons di kedai milik Kebin sudah lebih tertata. Tapi ada masalah baru, yaitu ternyata banyak Nimons yang hobi ngaku-ngaku. Contohnya, Pop dengan santainya mengambil 3 gorengan, tapi begitu ditagih malah ngakunya cuma makan 1 gorengan. Sementara itu, saldo *e-wallet* milik Toto tinggal 5 ribu, tapi masih nekat pesan Pisang goreng jumbo seharga 15 ribu, jadi pas bayar malah ngutang dulu.

Supaya bisnis F&B Kebin nggak bangkrut gara-gara Nimons nakal, ia butuh sistem yang bisa mencatat pembeli secara rapi, mengecek apakah saldo mereka cukup untuk membayar pesanan.

Petunjuk

1. Buat kelas bernama `Pembeli` yang berisi data nama pembeli, nomor meja, dan saldo e-walletnya.
2. Buat konstruktor yang menerima semua atribut sebagai parameter.
3. Buat getter dan setter untuk setiap atribut.
4. Buat method tambahan:
   - `cekBisaBayar(OrderMakanan order)` → mengecek apakah saldo cukup untuk membayar total harga OrderMakanan.
5. Lengkapi **`Pembeli.java`** dan gabungkan file `Pembeli.java` dan `OrderMakanan.java` menjadi zip dengan nama `Pembeli.zip`.

Java 8

Pembeli.zip

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	12	Accepted	0.18 sec, 28.20 MB
2	12	Accepted	0.40 sec, 27.94 MB
3	12	Accepted	0.67 sec, 28.50 MB
4	12	Accepted	0.40 sec, 27.98 MB
5	12	Accepted	0.40 sec, 28.03 MB
6	12	Accepted	0.41 sec, 27.96 MB
7	12	Accepted	0.19 sec, 27.86 MB
8	16	Accepted	0.17 sec, 28.11 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: WarungKebin.java

Setelah sistem `OrderMakanan` dan `Pembeli` yang Anda buat berjalan, bisnis F&B milik Kebin semakin ramai oleh Nimons yang datang. Kini, Kebin ingin tahu "apakah setiap pembeli mampu membayar pesanannya" secara teratur.

Sebagai asisten Kebin, Anda diminta melengkapi sebuah program utama (`WarungKebin.java`) yang:

1. Membaca berapa banyak pembeli yang datang
2. Membaca setiap data pembeli beserta pesanannya
3. Mengecek apakah setiap pembeli bisa membayar sendiri pesanannya.
   - Jika bisa, saldo pembeli berkurang sebesar total harga pesanan
   - Jika tidak, saldo tetap utuh
4. Mencetak laporan untuk setiap pembeli

Petunjuk

1. Gunakan class `OrderMakanan` (soal 1) dan `Pembeli` (soal 2).
2. Buat class main `WarungKebin` dengan method `main`.
3. Gunakan loop untuk membaca data semua pembeli dan menentukan hasil akhir.
4. Lengkapi dan kumpulkan file `WarungKebin.java` saja.

Batasan

1. Jumlah pembeli N ≥ 1.
2. Input dijamin valid sesuai format.

Format Masukan

- Satu bilangan bulat positif **N** (**N** ≥ 1), yang mewakili pembeli yang datang ke Warung Kebin
- Mengisi data pembeli (nama pembeli, nomor meja, dan saldo) dan data order (nama makanan, jumlah, harga satuan) sesuai format asli
- Ulangi sebanyak N kali

Format Keluaran

- Menampilkan laporan setiap pembeli sesuai input

Contoh Masukan dan Keluaran

No	Masukan	Keluaran	Keterangan
1.	1 Pop 3 50000.00 Bakso 2 20000.00	=== Pembeli 1 === Nama: Pop (Meja 3) Pesanan: Bakso x 2 @ 20000.0 Total: 40000.0 Bisa Bayar: Bisa Saldo setelah bayar: 10000.0	

2.	3 Toto Suroto 5 55000.00 Ayam Bakar 2 20000.00 Stewart 6 20000.00 Mie Ayam 2 15000.00 Daev Budiman 7 30000.00 Rawon 3 10000.00	=== Pembeli 1 === Nama: Toto Suroto (Meja 5) Pesanan: Ayam Bakar x 2 @ 20000.0 Total: 40000.0 Bisa Bayar: Bisa Saldo setelah bayar: 15000.0 === Pembeli 2 === Nama: Stewart (Meja 6) Pesanan: Mie Ayam x 2 @ 15000.0 Total: 30000.0 Bisa Bayar: Gak Bisa === Pembeli 3 === Nama: Daev Budiman (Meja 7) Pesanan: Rawon x 3 @ 10000.0 Total: 30000.0 Bisa Bayar: Bisa Saldo setelah bayar: 0.0	
3.	2 Jeffrey 4 40000.00 Bakso Urat 4 10000.00 Karel 6 25000.00 Nasi Goreng 1 25000.00	=== Pembeli 1 === Nama: Jeffrey (Meja 4) Pesanan: Bakso Urat x 4 @ 10000.0 Total: 40000.0 Bisa Bayar: Bisa Saldo setelah bayar: 0.0 === Pembeli 2 === Nama: Karel (Meja 6) Pesanan: Nasi Goreng x 1 @ 25000.0 Total: 25000.0 Bisa Bayar: Bisa Saldo setelah bayar: 0.0	

NOTE:

Pastikan setiap output diakhiri oleh **endline ("\n") atau println**! Khusus di line paling terakhir tetap dua kali endline seperti laporan pembeli yang lainnya

Java 8

WarungKebin.java

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.08 sec, 29.93 MB
2	10	Accepted	0.07 sec, 29.63 MB
3	10	Accepted	0.07 sec, 30.35 MB